UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/704,649	11/02/2000	Sinha Navin Kumar	JP920000155US1	4492

30449 7590 10/24/2005

SCHMEISER, OLSEN + WATTS
3 LEAR JET LANE
SUITE 201
LATHAM, NY 12110

EXAMINER	
CHANG, SUNRAY	

ART UNIT	PAPER NUMBER
2121	

DATE MAILED: 10/24/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/704,649	KUMAR, SINHA NAVIN	
	Examiner	Art Unit	
	Sunray Chang	2121	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 August 2005*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)□ Claim(s) _____ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-15,17,18 and 20* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This office action is in responsive to the paper filed on August 10th, 2005.

Claims 1 – 15, 17 – 18, and 20 are presented for examination.

Claims 1 – 15, 17 – 18, and 20 are rejected.

Claims 16 and 19 are cancelled.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

2. **Claims1 – 15, 17 – 18, and 20 are rejected** under 35 U.S.C. 103(a) as being unpatentable over Andrew E. Ayers (U.S. Patent No. 5,857,105, and referred to as **Ayers** hereinafter), and in view of Seema Hiranandani (U.S. Patent No. 5,812,855 and referred to as **Hiranandani** hereinafter).

(**Ayers** as set forth above generally discloses the basic invention.)

Regarding independent claims 1, 8 and 15,

Ayers teaches,

- A method for optimizing computer software [Col. 1, Lines 5 – 10] that includes one or more call statements [caller, Col. 2, Line 36] and a procedure [callee's routine, Col. 2, Line 37] which is callable by the or each call statement [caller, Col. 2, Line 36], the method being implemented by execution of program instructions by a computer, the program instruction being stored in computer-readable memory, [Fig. 1] the method comprising the steps of:

  (a) analyzing the callable procedure [comparing …callee routines, Col. 2, Lines 41 – 44], the callable procedure [callee routines] comprising a branch condition [direct, indirect call or in-line] under which control flow code [program listing] directs program flow [converts an indirect call] from the branch condition to a code branch of two or more code branches [employing call statistics associated with remaining prospective callee routines], each said code branch [callee routine] being within the callable procedure and branching from the branch condition to program code within the callable procedure [converts an indirect call to a callee routine in a caller routine program listing], said analyzing the procedure identifying the control flow code and the code branches [identifying]; [Col. 2, Lines 35 – 59]

  (b) identifying for each said code branch a new procedure containing the respective code branch and not containing the other code branches of the two or more code branches, wherein the new procedures collectively comprise the two or more code branches; [Col. 2, Lines 51 – 59, see also Lines 35 – 50]

(c) recording a list of data entries [source listing, Col. 1, Line 7] corresponding to the respective new procedures [replaces, Col. 1, Line 10], each entry comprising a data item identifying the respective new procedure [executable object code, Col. 1, Line 9] and a data item representative of the branch conditions [indirect/ direct, Col. 2, Lines 37 – 38] under which said control flow code [caller's routine, Col. 2, Line 43] directs program flow to the associated code branch [callee, Col. 2, Line 43]; [see also Col. 1, Lines 5 – 10, and Col. 2, Lines 35 – 59]

(d) for the or each call statement [caller, Col. 2, Line 36], scanning the entries in said list [source listing, Col. 1, Line 7] to determine one for which there is correspondence [a match, Col. 2, Line 44] between said branch conditions [indirect/ direct, Col. 2, Line 37 – 38] and call parameters [characteristics, Col. 2, Line 44] directed to said control flow code [caller's routine, Col. 2, Line 43] by the call statement [caller, Col. 2, Line 36] and modifying the call statement [reduces a number of indirect calls, Col. 1, Line 8 – 9] to replace [replaces, Col. 1, Line 10] the call to the original procedure by a call to the corresponding new procedure [in-line listing/direct calls, Col. 1, Line 10]. [see also Col. 1, Lines 5 – 10, Col. 2, Lines 35 – 59, Col. 3, Lines 22 – 50, and Fig. 2]

Ayers does not teach for each branching node, two or more further nodes representing respective code branches to which program flow is directed by the branching node.

Hiranandani teaches for each branching node, two or more further nodes representing respective code branches to which program flow is directed by the branching node [see Fig. 5, Col. 1, Lines 1 – 67], for the purpose of performing optimization on procedures. [see Col. 11, Lines 59 – 60].

It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teaching of **Ayers** to include "for each branching node, two or more further nodes representing respective code branches to which program flow is directed by the branching node", for the purpose of performing optimization on procedures. [see **Hiranandani**, Col. 11, Lines 59 – 60].

Regarding dependent claims 2 and 9,

Ayers teaches,

- constructing a control flow graph [constructs a global call graph, Col. 4, Line 19 – 20] for the procedure [transform procedure, Col. 4, Line 19], [Col. 4, Lines 10 – 20] the control flow graph comprising
- a branching node [callee node, Fig. 3 – 5, and Col. 4, Line 40] representative of said control flow code [caller's routine, Col. 4, Line 53] and further nodes representing respective ones of said code branches [callee node, Col. 4, Line 53]. [see also Col. 4, Lines 38 – 65, and Fig. 3 – 5]

Regarding dependent claims 3 and 10,

Ayers teaches,

- for each item of control flow code, before identifying any new procedure in accordance with step (b) of the method, checking for compliance between one or more predetermined rules for the software and the software should step (b) and following steps of the method take place; and for that item of control flow code, continuing with step (b) and the following steps of the method only in the event of such compliance. [signature match test, Col. 4, Line 66 – Col. 5, Line 18, see rejections of claim 1 for details]

Ayers does not teach a code branches itself comprises two or more code branches.

Hiranandani teaches a code branches itself comprises two or more code branches [see Fig. 5, see also Col. 1, Lines 1 – 67], for the purpose of performing optimization on procedures. [see Col. 11, Lines 59 – 60]

It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teaching of **Ayers** to include "a code branches itself comprises two or more code branches" for the purpose of performing optimization on procedures. [see **Hiranandani**, Col. 11, Lines 59 – 60]

Regarding dependent claims 5 and 12,

Ayers teaches,

- including optimizing the or each new procedure [reduces, Col. 5, Line 28] for which a call parameter is a constant [predetermined number, Col. 5, Line 27] by propagating that constant through the new procedure. [see also Col. 4, Line 51 – Col. 5, Line 59]

Regarding dependent claims 6 and 13,

Ayers teaches,

- including analyzing [comparing, Col. 2, Line 41] a call statement [caller, Col. 2, Line 43], [see also Col. 2, Lines 41 – 48]

- calling parameters [characteristic, Col. 2, Line 44] and an associated new procedure [callee routings, Col. 2, Line 38] to determine if they are compliant with predetermined in-lining rules [in-lining, Col. 2, Lines 53 – 55] and, [see Col. 2, Lines 35 – 58]
- if they are so compliant, replacing said call statement by a copy of the new procedure. [see Col. 1, Lines 5 – 10]

Regarding independent claims 7, 14, 17 – 18 and 20,

Ayers teaches,

- A method for optimizing computer software that includes one or more call statements and a procedure which is callable by the or each call statement, [see Col. 1, Lines 5 – 10, Col. 2, Lines 35 – 59, see also Col. 3, Lines 22 – 50, and Fig. 2], the method being implemented by execution of program instructions by a computer, the program instructions being stored in a computer-readable memory, [Fig. 1] the method comprising the steps of:
- constructing a control flow graph for the callable procedure, the control flow graph comprising one or more branching nodes each representative of respective control flow code and, [see Col. 4, Lines 38 – 65, Fig. 3 – 5, and Col. 4, Lines 10 – 20, see rejections of claim 2 for further detail], wherein said one or more branching nodes and said respective code branches are contained within the callable procedure; [Col. 2, Lines 35 – 59]
- considering each node in turn [sequentially accessed, Col. 4, Line 54] and, if the node being considered is a branching node and if the branching condition for that node by which the respective control flow code directs program flow to the respective code branches is able to be represented as a function only of formal parameters [signature, Col. 5, Line 21] and global

variables [Fig. 2], identifying a new procedure for which the flow control graph comprises all the nodes in the path from the first node of the procedure to the node being considered, the node being considered, and the whole of the portion of the control flow graph led to directly or indirectly [indirect/ direct, Col. 2, Lines 37 – 38] from the node being considered; [see Col. 4, Lines 51 – 65, Col. 5, Lines 10 – 31, and Col. 4, Lines 66 – Col. 5, Lines 9]

- recording a list of data entries [source listing, Col. 1, Line 7] corresponding to the respective new procedures, each entry comprising a data item identifying the respective new procedure and a data item representative of the corresponding branching condition; [Col. 1, Lines 5 – 10, and Col. 2, Lines 35 – 59]
- for each said call statement, scanning the entries in said list to determine one for which there is correspondence between said branch condition and call parameters supplied by the call statement; and modifying the call statements to call said new procedures. [see Col. 1, Lines 5 – 10, Col. 2, Lines 35 – 59, see also Col. 3, Lines 22 – 50, and Fig. 2]
- implemented in the form of a body of computer code made available for downloading from a computer connected to a computer network. [Fig. 1]

Ayers does not teach for each branching node, two or more further nodes each connected to said each branching node by respective code branches to which program flow is directed from the branching node.

Hiranandani teaches for each branching node, two or more further nodes each connected to said each branching node by respective code branches to which program flow is directed from the branching node [see Fig. 5, Col. 1, Lines 1 – 67], for the purpose of performing optimization on procedures. [see Col. 11, Lines 59 – 60].

3. **Claims 4 and 11 are rejected** under 35 U.S.C. 103(a) as being unpatentable over **Ayers**, and in view of Seema Hiranandani, and further in view of Uma Mahadevan (U.S. Patent No. 5,797,013 and referred to as **Mahadevan** hereinafter).

Regarding dependent claims 4 and 11,

Ayers teaches,

- A method for optimizing computer software that includes one or more call statements and a procedure which is callable by the or each call statement, [see Col. 1, Lines 5 – 10, Col. 2, Lines 35 – 59, see also Col. 3, Lines 22 – 50, and Fig. 2]

Ayers does not teach the application of a cost-analysis algorithm based on predetermined rules about the length of the software.

Hiranandani teaches a code branches itself comprises two or more code branches [see Fig. 5, and Col. 1, Lines 1 – 67], for the purpose of performing optimization on procedures.

Mahadevan teaches the application of a cost-analysis algorithm based on predetermined rules about the length of the software [Col. 10, Lines 46 – 52 & Lines 7 – 45] for the purpose of optimization.

It would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teaching of **Ayers** to include "the application of a cost-analysis algorithm based on predetermined rules about the length of the software" for the purpose of optimization. [see **Hiranandani**, Col. 10, Lines 46 – 52]

Response to Amendment

Claim Rejections - 35 USC § 101

4. Applicants' amendment overcomes claims to avoid 35 USC § 101 rejections. The rejections based on references have been withdrawn.

Claim Rejections - 35 USC § 102

5. Applicants' amendment modified rejected claims to overcome 35 USC § 102 rejections. The 102 rejections based on reference **Ayers** have been withdrawn. Yet, further 35 USC § 103 rejections have been cited in current office action for newly modified claims.

6. Examiner agrees applicants' argument regarding "**Ayers** does not teach that the callee routine comprises a branch condition under which control flow code directs program flow from the branch condition to a code branch of two or more code branches". Yet, **Ayers** reference is not used for rejecting these subject matter, **Hiranandani** reference was cited in former office action and also in current office action for rejecting these related subject matter.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sunray Chang whose telephone number is (571) 272-3682. The examiner can normally be reached on M-F 7:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Anthony Knight can be reached on (571) 272-3687. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-746-3506.

Sunray Chang
Patent Examiner
Group Art Unit 2121
Technology Center 2100
U.S. Patent and Trademark Office

October 18, 2005

Anthony Knight
Supervisory Patent Examiner
Group 3600